

WHAT IS CLAIMED IS:

1. A method for stack memory protection comprising the steps of:
- generating new memory page attributes for a page table used to manage memory, each of said new memory page attributes identifying a block of memory as a new class of memory, each of said new memory page attributes generated by a corresponding new load/store instruction;
 - assigning, by an operating system or a processor, a selected one of said new memory page attributes to a selected block of memory, said selected block of memory used as a new class of memory corresponding to said selected new memory page attribute;
 - blocking normal load /stores to a memory block having one of said new memory page attributes; and
 - blocking a first new load/store to a memory block with one of said new memory page attributes not corresponding to said first new load/store

1 2. The method of claim 1, wherein said new classes of memory comprise stack
2 memory.

1 3. The method in claim 2, wherein a first error condition is generated whenever
2 normal load/stores are attempted to stack memory having a first or a second stack
3 memory attribute.

1 4. The method in claim 2, wherein a second error condition is generated
2 whenever said stack memory load/stores are attempted to memory not having said
3 stack memory attribute.

1 5. The method in claim 2, wherein a third error condition is generated whenever
2 a stack memory load/store for a first memory stack is attempted to a second memory
3 stack, said third error condition also generated if a stack memory load/store for said
4 second memory stack is attempted to said first memory stack.

1 6. The method of claim 2, wherein said stack memory load/store
2 instructions are executed on a CPU comprising an IA64 architecture.

1 7. The method of claim 5, wherein said first memory stack is a processor stack,
2 said processor stack used by a processor to load and store hardware register contents

3 during program execution, said processor stacks transparent to a programmer or a
4 compiler.

1 8. The method of claim 7, wherein said processor stack is an IA64 register stack.

1 9. The method of claim 5, wherein said second memory stack is a program stack,
2 said program stack used by a programmer or a compiler in managing program flow.

004060 TET 45960

1 10. A processor comprising stack memory protection circuitry, said processor
2 using blocks of memory as stack memory, said stack memory protection circuitry
3 comprising:

4 a stack memory attribute circuit, said stack memory attribute circuit
5 operable to generate memory attributes, said memory attributes
6 associated with each memory block designated as a memory stack;

7 a page table attribute storage circuit, said page table attribute circuit
8 operable to store and associate one of said stack memory attributes
9 with a block of memory designated as stack memory;

10 a stack memory allocation circuit, said stack memory allocation circuit
11 operable to identify a block of memory as a stack memory and
12 associate said memory block with one of said stack memory attributes,
13 said stack memory attributes stored in a memory page table; and

14 a stack memory instruction execution circuit, said stack memory
15 instruction execution circuit operable to decode load/store instructions
16 to memory blocks, said stack memory instruction execution circuit
17 granting stack memory load and stores to memory blocks having a

002050-121-050700

09571-090700

1 15. The processor of claim 13, wherein said first memory stack is a processor
2 stack, said processor stack used by a processor to load and store hardware register

3 contents during program execution, said processor stacks transparent to a programmer
4 or a compiler.

1 16. The processor of claim 13, wherein said second memory stack is a program
2 stack, said program stack used by a programmer or a compiler in managing program
3 flow.

09657121-090700

1 17. A data processing system, comprising:

2 a central processing unit (CPU);
3 shared random access memory (RAM);
4 read only memory (ROM);
5 an I/O adapter; and
6 a bus system coupling said CPU to said ROM, said RAM said display
7 adapter, wherein said CPU, said CPU comprising stack memory
8 protection circuitry, said stack memory protection circuitry
9 comprising:

10 a stack memory attribute circuit, said stack memory attribute circuit
11 operable to generate memory attribute, said memory attribute
12 associated with each memory block designated as a memory stack;

13 a page table attribute storage circuit, said page table attribute circuit
14 operable to store and associate said stack memory attribute with a
15 block of memory designated as stack memory;

16 a stack memory allocation circuit, said stack memory allocation circuit

007060" 121.090700
00657.121.09060

operable to identify a block of memory as a stack memory and
associate said memory block with a stack memory attribute, said stack
memory attribute stored in a memory page table; and

a stack memory instruction execution circuit, said stack memory
instruction execution circuit operable to decode load/store instructions
to memory blocks, said stack memory instruction execution circuit
granting stack memory load and stores to memory blocks having a
stack memory attribute and not granting stack memory load and stores
to memory blocks not having said stack memory attribute.

18. The data processing system in claim 17, wherein a first error condition is
generated whenever normal load/stores are attempted to stack memory having a first
or a second stack memory attribute.

19. The data processing system in claim 17, wherein a second error condition is
generated whenever said stack memory load/stores are attempted to memory not
having a stack memory attribute.

20. The data processing system in claim 17, wherein a third error condition is
generated whenever a stack memory load/store for a first memory stack is attempted

095713-090700

Figure 1

[illegible]

	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100
1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	

1 24. A computer program product embodied in a machine readable medium,
2 including an operating system and a compiler for a processor system, comprising; a
3 program of instructions for performing the program steps of:

4 generating new memory page attributes for a page table used to
5 manage memory, each of said new memory page attributes identifying
6 a block of memory as a new class of memory, each of said new
7 memory page attributes generated by a corresponding new load/store
8 instruction;

9 assigning, by an operating system or a processor, a selected one of said
10 new memory page attributes to a selected block of memory, said
11 selected block of memory used as a new class of memory
12 corresponding to said selected new memory page attribute;

13 blocking normal load /stores to a memory block having one of said
14 new memory page attributes; and

15 blocking a first new load/store to a memory block with one of said new
16 memory page attributes not corresponding to said first new load/store

00/05/01 12:55:00

30. The computer program product of claim 29, wherein said first memory stack is a processor stack, said processor stack used by a processor to load and store hardware

3 register contents during program execution, said processor stacks transparent to a
4 programmer or a compiler.

1 31. The computer program product of claim 30, wherein said processor stack is an
2 IA64 register stack.

1 32. The computer program product of claim 28, wherein said second memory
2 stack is a program stack, said program stack used by a programmer or a compiler in
3 managing program flow.

002050" PAT 25960

1 33. A method of managing a memory device comprising the steps of:

2 partitioning said memory device into a plurality of memory spaces on
3 an as-needed basis; and

4 associating a memory attribute with each memory space; said memory
5 attribute determining a use of each of said memory spaces.

1 34. The method of claim 33, wherein a particular memory attribute has
2 corresponding load/store instruction.

1 35. The method of claim 34, wherein a load/store instruction associated with a
2 first memory attribute causes an error condition if attempted on a memory space with
3 a second memory attribute.

1 36. The method of claim 33, wherein each of said memory attributes are stored in
2 a memory page table, said memory page table used to manage said memory device.